

OBSERVACIONES DEL LA PRACTICA

María Alejandra Moreno Bustillo Cod 202021603

Juliana Delgadillo Cheyne Cod 202020986

Preguntas de análisis

a) ¿Qué instrucción se usa para cambiar el límite de recursión de Python?

Para cambiar el límite de recursión en python se usa la instrucción ilustrada en la linea 158 "sys.setrecursionlimit()".

```
156 if __name__ == "__main__":
157     threading.stack_size(67108864) # 64MB stack
158     sys.setrecursionlimit(2 ** 20)
159     thread = threading.Thread(target=thread_cycle)
160     thread.start()
161
```

b) ¿Por qué considera que se debe hacer este cambio?

El propósito de cambiar este límite es para que se pueda hacer un mayor número de recursiones y se puedan llevar a cabo los algoritmos necesarios en los gráficos que funcionan mediante profundidad, esto implica un gran número de recursiones, especialmente cuando se trabaje con conjuntos de datos grandes y grafos densos, lo cual haremos en este caso. Es por esto, que el limite de recursiones predeterminados por Python no es suficiente.

c) ¿Cuál es el valor inicial que tiene Python cómo límite de recursión?

Como límite de recursión Python maneja 1000 llamadas recursivas como valor default. Cabe mencionar que, desde la programación, este valor puede ser modificado según el programa que se desee ejecutar.

d) ¿Qué relación creen que existe entre el número de vértices, arcos y el tiempo que toma la operación 4?

Archivo	Número de Vértices	Número de Arcos	Tiempo
bus_routes_50.csv	74	73	48.771000000000001
bus_routes_150.csv	146	146	63.395999999999995
bus_routes_300.csv	295	382	115.960000000000001
bus_routes_1000.csv	984	1633	559.255000000000001
bus_routes_2000.csv	1954	3560	1046.875
bus_routes_3000.csv	2922	5773	2109.375
bus_routes_7000.csv	6829	15334	6687.5
bus_routes_10000.csv	9767	22758	21781.25
bus_routes_14000.csv	13535	32270	30828.125

Como se evidencia en la tabla, a medida que aumenta la cantidad de datos, se crean mayor numero de vértices y arcon, lo que lleva a un mayor tiempo de ejecución. Lo anterior tiene sentido, porque el grafo esta aumentando su tamaño. Lo mismo ocurre al ejecutar la opción 6, a medida que aumentan los datos, los tiempos de ejecución aumentan igualmente.

Archivo	Número de Vértices	Número de Arcos	Tiempo
bus_routes_50.csv	74	73	1.1859999999999649
bus_routes_150.csv	146	146	1.2360000000000149
bus_routes_300.csv	295	382	1.2610000000000121
bus_routes_1000.csv	984	1633	1.327999999999959
bus_routes_2000.csv	1954	3560	1.43900000000004122
bus_routes_3000.csv	2922	5773	1.6050000000000786
bus_routes_7000.csv	6829	15334	1.61800000000005634
bus_routes_10000.csv	9767	22758	2.058999999999145
bus_routes_14000.csv	13535	32270	2.1160000000000895

- e) ¿El grafo definido es denso o disperso?, ¿El grafo es dirigido o no dirigido?, ¿El grafo está fuertemente conectado?

El grafo definido es disperso y esto se puede comprobar mediante la siguiente expresión: $e/v(v-1)$, donde e es el número de arcos y v es el número de vértices, cuando reemplazo los valores con los recibidos en el archivo más grande, obtengo que la densidad del grafo es de 0,00018 lo cual es una densidad bastante baja. Por otro lado, el grafo es dirigido debido a que en la creación se establece que el parámetro directed sea verdadero. Por último, el grafo está fuertemente conectado puesto que el número de arcos en el archivo más grande es más del doble del número de vértices presentes, por lo que hay muchas conexiones creadas entre estos vértices.

```
analyzer['connections'] = gr.newGraph(datastructure='ADJ_LIST',
                                     directed=True,
                                     size=14000,
                                     comparefunction=compareStopIds)
return analyzer
```

- f) ¿Cuál es el tamaño inicial del grafo?

El tamaño inicial del grafo es de 14000.

- g) ¿Cuál es la Estructura de datos utilizada?

La estructura de datos utilizada es de lista adyacente o ADJ list.

- h) ¿Cuál es la función de comparación utilizada?

La función de comparación utilizada es llamada “compareStopIds” y con esta se podrán hacer las comparaciones necesarias para generar las conexiones en los grafos.